View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Dagstuhl Research Online Publication Server

Petri Net Modelling for Enhanced IT Asset Recycling Solutions

Christina Latsou[1], Sarah J. Dunnett[2], and Lisa M. Jackson[3]

1 Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK
C.Latsou@lboro.ac.uk
2 Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK
3 Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK

Abstract

From preliminary design through product sustainment to end of life removal, optimal performance through the entire life cycle, is one of the most important design considerations in engineering systems. There are a number of mathematical modelling techniques available to determine the performance of any system, or process design. This paper focuses on the Petri Net technique for the representation and simulation of complex cases with the future aim of automatically generating a model from the system, or process description. If the model can be automatically generated changes can be investigated easily, enabling different designs to be investigated. Within this research, a Petri Net model is developed for a process of recycling IT assets. The model developed here will be used in future work to validate the automation process. This model is simulated and programmed in Matlab. The model enables the simulation of various flow paths through the recycling process, giving an understanding of the current process limiting factors. These can then be used to identify possible ways of improving the efficiency of the recycling process and enhancing the current IT asset management strategy. The future aim of this research is the automatic generation of a system model for complex industrial systems and processes by converting the SysML-based specifications into Petri Nets.

1998 ACM Subject Classification D.2.2 Design Tools and Techniques, I.6 Simulation and Modelling

Keywords and phrases Petri Net, simulation, modelling, asset management

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.3

1 Introduction

At the earliest stages of the design of a system, or process, it is essential to consider its performance in order to ensure that the optimal design is developed. A number of modelling techniques are available, facilitating the performance evaluation and analysis of any system, or process. Once the performance is understood, the decision maker is able to make informed decisions about the design. The techniques can be divided into two main categories, analytical and simulation techniques. The analytical techniques include a wide variety of approaches. An analyst is able to choose the most suitable technique, given the available data and the preferred analysis that needs to be performed. The techniques for failure analysis consist of combinatorial models, including Reliability Block Diagrams (RBDs), Fault Trees (FTs) and Binary Decision Diagrams (BDDs), state-space models, including the subcategory of Markov approaches, and hierarchical models generated by the combination of combinatorial

© Christina Latsou, Sarah J. Dunnett, and Lisa M. Jackson;
licensed under Creative Commons License CC-BY
5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza; Article No. 3; pp. 3:1–3:10
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and state-space models, which are able to simplify the model and ease further analysis [1]. In the case of investigating other performance measures, different analytical techniques can be used. The models mentioned above contain certain limitations rendering them not applicable to some cases, especially when the system, or process, includes complex structures. More specifically, the combinatorial models are not able to model dynamic characteristics, such as dependent events and spares, whereas the Markov models, which can cope with dynamic features, suffer the state-space explosion as the number of states increases. In other words, large complex scenarios are difficult to be modelled and controlled using Markov approaches, as the final diagram can be very large, difficult to be built due to its complexity and computationally costly.

However, alternative approaches have been developed, such as the encoding of the state-space model in a Petri Net (PN) that can cope with all the aforementioned limitations [7]. Petri Net models are powerful, flexible structures that can be applied to complex cases without suffering the state-space explosion limitation.

The main characteristics of a complex case are redundancy, configurability and interactions between systems components or processes scenarios. These features are usually incorporated into the system, or process creating high level of difficulties in the manual development process of performance models. Complex cases can be recognized by the following factors: the system size (number of different components, or cases within it), the system dynamics i.e. the extent to which behaviour changes over time, the fault and error recovery, the sequence-dependent failures, the use of spares and the existence of phased missions [4].

Nowadays, the use of computational models is considered an essential element for the analysis of complex cases. This paper uses the simulation technique which has become a ubiquitous tool for the analysis and investigation of complex cases. It is characterised as a straightforward, accurate, easily applicable and flexible method that can represent the system, or process in an efficient way and hence it can provide reusability of the model for different designs and policies, enabling informed decisions to be made. Hence, in this paper a simulation method is generated with the modelling power of Petri Nets to model a process of recycling IT assets. The results from the simulation of the various flow paths of the process give the opportunity for the analyst to detect the limiting factors in the process and make the appropriate decisions. As a result, the efficiency of the current process can be improved and the IT asset management strategy can be enhanced.

The future aim of the project is development of a general methodology for complex cases, implemented in computer software, which will accept as input a system, or process description and automatically generate the corresponding Petri Net model. The study developed here is the first step in this process, the manual development of a PN. The future stages will develop an algorithm to do this automatically for the process described. This will be generalised for any complex system, or process. This paper is organised as follows. Section 2 describes briefly the PNs giving a general overview of basic concepts and elements of the models. In Section 3, a PN model is developed for a process of recycling IT assets and results are obtained from the net highlighting certain aspects of the process. Some general conclusions are drawn in Section 4.

2 Petri Nets

First introduced in the thesis of C. A. Petri in 1962 [5], Petri Net is a visual tool that provides rigorous and precise model analysis. PNs have been applied to a wide spectrum of cases in different sectors, such as data communication processes, computer networks, workflows and

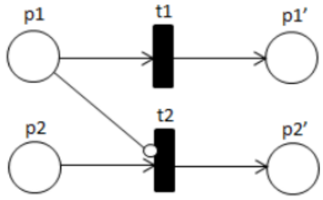

Figure 1 Inhibitor Arc.

manufacturing plants [6]. A Petri Net is a bipartite directed graph that includes two types of nodes: places, drawn as circles, and transitions, drawn as bars. There are two types of transitions, the immediate (drawn as solid rectangles) which when enabled fire immediately, and timed (drawn as hollow bars) which have a time delay associated with them. Directed edges (arcs) connect places to transitions and vice versa. The dynamic behaviour of the model is described by the movement of tokens/markers (solid dots) between places. A token can move between places only if the corresponding transition has been enabled. A transition is said to be enabled and able to fire if the number of tokens in each of its input places is at least equal to the multiplicity of the corresponding arc from that place. The multiplicity of an arc is described as the number of the tokens that are removed from the place during a firing transition. This multiplicity is denoted by using a slash through the arc a positive integer beside it. When the multiplicity is equal to 1, then the slash and integer can be omitted. Another element that was added in order to increase the decision power of the Petri Nets is the inhibitor arc, denoted as an arc terminated with a hollow circle. The inhibitor arc prevents the firing of a transition when the place it comes from is marked. According to Figure 1 t1 is enabled if p1 contains a token, while t2 is enabled if p2 contains a token and p1 has no token.

The movement of tokens through a Petri Net can be transformed into matrix form. Then the marking of the PN after the $_r$ transition, M_r, can be found by Equation (1).

$$M_r = M_0 + A^T \cdot T_1 \,. \tag{1}$$

Where: M_0 is a column matrix $(n, 1)$, where n is the number of places, showing the initial marking of the net. T_1 is a column matrix $(m, 1)$ where m is the number of transitions, showing the number of times each transition has fired in the r transitions. A is the incidence matrix (m, n) where each element a_{ij} corresponds to the effect that transition i has on place j. Using Equation (1), the marking of a net, the distribution of tokens within it, can be determined at any time.

Petri Net models can predict the performance of complex processes, being a powerful modelling tool suitable for the integration of continuous and discrete dynamics in the model. According to Ling [2] reachability, boundedness and liveness are three main behaviour properties that relate to each other and can be identified in PN models. Reachability is the capability to reach one particular state (M_0) from another (M_0') if there is a sequence of transitions such that $M_0 \leq M_0'$. A Petri Net is characterised as bounded if the number of tokens in each place is less or equal to a finite number k for any marking reachable from the initial marking. The term liveness ensures that a transition is live if it can never deadlock.

In the area of applied mathematics Petri Nets are of special interest since they can be analysed, simulated and modelled numerically and graphically, using various software tools, such as C++, Matlab/Simulink, PN Toolbox [3], SimHPN (GISED) and their tools.

In this application, given the IT asset process, a Petri Net model is developed. The novelty of application in this domain for asset management of recycling is the ability to investigate and identify possible limiting factors in the process and hence optimise the performance of the IT process.

3 Simulation

The Petri Net technique, described above, can be used for the simulation of a process in order to conduct performance analysis. To demonstrate the technique of simulation, a PN model has been developed for a process of recycling IT assets. The model enables the simulation of various flow paths through the recycling process. The simulation results will give the ability to the analyst to identify possible ways of improving the performance of the process and enhancing the current IT asset management strategy.

3.1 Process Description

A recycling IT process has been considered and modelled using Petri Nets in order to simulate its performance. The process focuses on the repair of electronic devices, primarily mobiles phones. The product enters in the process line and then it can pass along one of the two paths, either refurbished, through the reuse line, or scrap. Considering the refurbished path, there are seven different possible stages:

- Asset Track (AT): Asset information is introduced into the traceability system.
- Visual Inspection (VI): The physical condition of an asset is assessed.
- Functional Test (FT): A product is inspected by testing its functionality, including activities such as charger check, battery test, LCD screen check, resetting, ringing test, vibration, microphone and speaker test.
- Data Erasure (DE): Data is erased securely by using specific licensed software.
- Cleaning and De-Labelling (C&D): Refurbished assets are cleaned and any non-essential labels are removed from the device. A new label is placed.
- Repair (R): A product can be repaired only if its repair is considered economically viable.
- Strip and Scrap (S&S): Failed products are checked for any parts that can be salvaged and are then sent for secure destruction.

In the case of a scrapped device, there are two options for it. It can be used either as a unit level, meaning price sought per tonne for scrap, or as a component level, meaning components are extracted from the device and used within this process for future repairs or sold for spares. All stages can only have one device at a time apart from the Data Erasure stage. Each stage has a time to completion associated with it, which can vary for different devices. Additionally, at some of the stage there is a probability of pass/ fail, according to the process. In practise, most of the activities are performed at the same physical location, i.e. on the computer. The repair stage (R) takes place in the same factory, but it is performed when there are a batch requiring repair and it takes place away from the main refurbishment process and for that reason there is a large delay between the functional test and the repair activities. Similarly, the data erasure stage is performed separately and then the information is only logged in the process at the end. The company's manpower is assumed to be 2.

The process diagram is presented in Figure 2, including all the possible paths of the recycling IT process.

3.2 Model Construction

From the information given, as described in Section 3.1, the corresponding Petri Net model has been developed and presented in Figure 3. The PN model includes all the stages referred

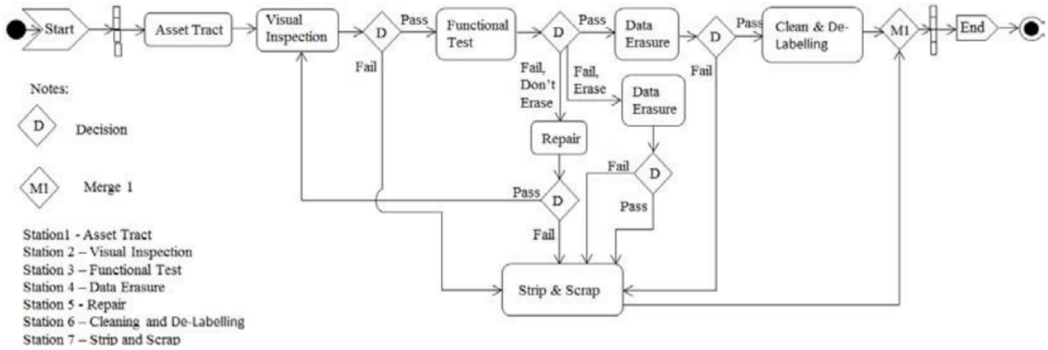

Figure 2 Recycling IT Process Diagram.

to in the process description. Two places and a timed transition have been used to represent each activity, i.e. one place is used to describe that the activity starts, whereas a second place is used to describe that the activity ends. The time for the activity to be completed is described by a timed transition; these are denoted as Tx, where x is a number (1–8), in Figure 3. It can be seen that there is a TI node which is used as a merge place only if two or more devices arrive at the same place. Inhibitors have been placed in the PN, as shown in Figure 3, allowing the firing of a transition only if the subsequent place is empty. This models the queuing of items. There is an immediate transition in the model, named TI and represented by a black box, which corresponds to an instantaneous transition.

Some stages of the process have a probability associated with their pass or fail and these are shown by probability transitions in Figure 3 denoted by Px, where x is a number (1–14). For example, at the start of the process after asset tracking has taken place a visual inspection occurs. This takes a time T2 and there is a probability P3 that the unit passes this inspection, etc. In order to model the time between stages in the model, places Pix, with associated times Tix, have been added. Where, as before, x is a number (1–33). So, in total the PN includes 29 places, 1 immediate transition and 33 timed transitions from which 8 correspond to the activities, 14 represent the intermediate time from one activity to another and the rest 11 transitions express the probabilities (pass/fail) of each activity.

3.3 Process Data and Model Simulation

In order to simulate the process modelled in Figure 3 it is necessary to input values for the timed and probabilistic transitions. Table 1 presents the minimum and maximum times needed to complete each activity and Table 2 shows the same data for the times between stages. The probabilities of pass (P) or fail (F) for each stage of the process are also listed in Table 2. The retrieved data, which was obtained from the IT asset management company, comes from 2113 mobile phones which have been processed for the time period of 323 hours.

Hence in the simulation, times are generated for the timed transitions by assuming that they all follow a continuous uniform distribution. For each transition a random number, x, is generated and the time obtained from Equation (2).

$$t = \min_t + [(\max_t - \min_t) \cdot x]\,. \tag{2}$$

Where:

- $\min_t$ and $\max_t$ are obtained from Tables 1 and 2 for each activity type and correspond to Min Time and Max Time respectively.
- x returns a uniformly distributed random number in the interval $(0, 1)$.

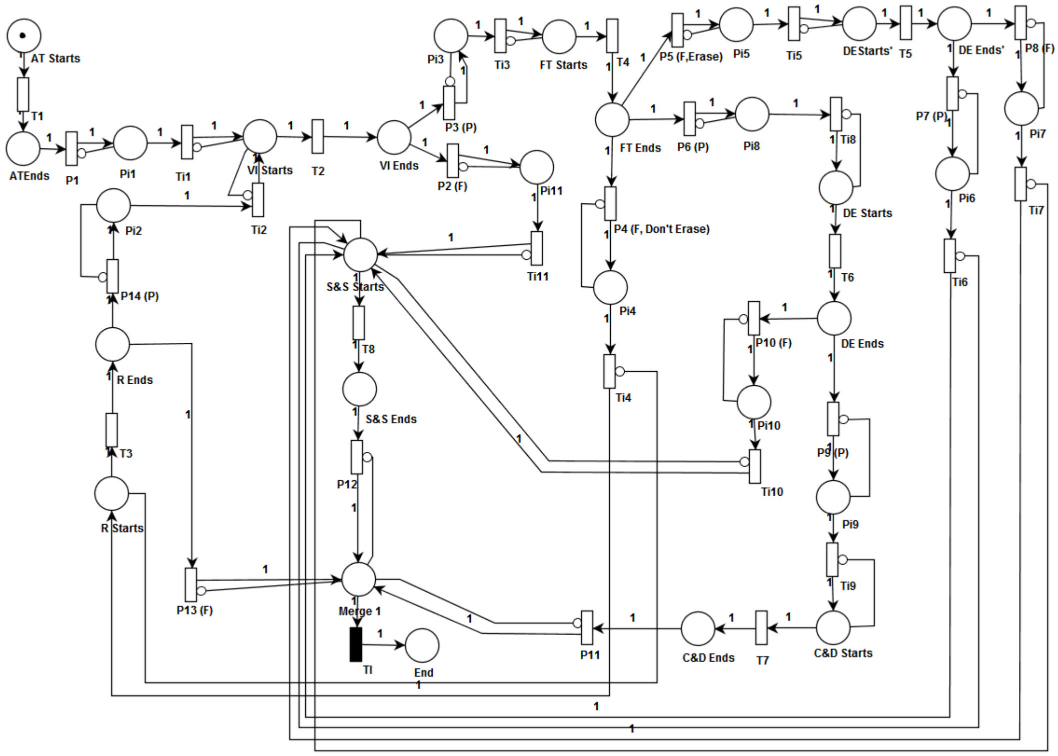

Figure 3 Recycling IT Process Petri Net Model.

The main steps followed for the process simulation are outlined in Figure 4. The Matlab code was generated and run for 1000 simulations. As mentioned, random numbers for the process times and probabilities are generated and times for activities and intermediate stages are estimated. Once a random probability is lower than or equal to the pass probability of an activity then the device is assumed to pass, otherwise the device fails. Following Figure 3 and the probabilities for each path the simulation results can be found. The simulation can provide the total number of repaired and scrapped devices, the average time for each activity and intermediate stage, the longest queues once more than one device exists in the process, as well as the most visited places and the most failed nodes in each path.

3.4 Results

This section presents the results obtained from the simulation of the PN developed in Section 3.2. The main focus of this application is the investigation of the process and identification of its limitations with the aim to find ways to enhance the performance of the process.

Table 3 presents the average times taken for the transitions included in the repair path for the 1000 simulations undertaken. It is observed that Ti4, which is the time needed from the end of the FT to the start of the R activity, is the longest time in this path. This happens because the r activity takes place in a different location from the rest of the activities and extra time is required for the transportation of the devices. As can be seen that the average times obtained agree well with the times given in Tables 1 and 2. According to the 6 paths identified in the PN and Table 5, the most common failed nodes have been found in each path

Table 1 Average Times Table for Process Stages.

Timed Transitions	Activity Type	Min Time (seconds)	Max Time (seconds)
T1	Asset Track	107	148
T2	Visual Inspection	5	10
T3	Repair	240	900
T4	Functional Test	60	180
T5, T6	Data Erasure	30	40
T7	Cleaning and De-Labelling	30	60
T8	Strip and Scrap	30	60

Table 2 Times between Stages and Probabilities for the IT Process.

Transition ID	Min Time (seconds)	Max Time (seconds)	Transition ID	Probability
Ti1	30	120	P1,11,12(P)	1
Ti2	1800	28800	P2(F)	0.312
Ti3	300	1800	P3(P)	0.688
Ti4	7200	86400	P4,5(F)	0.13365
Ti5	1800	7200	P6(P)	0.7327
Ti6	1800	10800	P7,9(P)	0.9702
Ti7	1800	10800	P8,10(F)	0.0298
Ti8	1800	7200	P13(F)	0.7066
Ti9	1800	10800	P14(P)	0.2934
Ti10	1800	10800		
Ti11	300	3600		

and presented in Table 4. The initial total number of runs is equal to 1000 and the results for the places depend on the probabilities of their equivalent transitions. The most common failed place to have been visited, Pi11, corresponds to VI failing and has a failure probability equal to 0.312. In addition, the most common visited places have been investigated in each path and it was found that the activities with the most visits are the AT and VI passing nodes. More specifically, the AT nodes are always visited, whereas the VI passing places are visited approximately at 68% in all paths.

Table 6 includes the overall average times for all PN paths for two different cases; the first case investigates only one device in the process, whereas the second case investigates two. For the estimation of the average repair time in the case of one device, the activity and intermediate times have been taken into consideration. In the case of two devices the methodology presented in Figure 5 has been followed. All the activities in the repair path can have only one device at a time, meaning that the 2^{nd} device can proceed with the AT, once the 1^{st} phone ends with the AT. The same happens for the R. Secondly, an additional time that should be included in the case of two mobile phones is the time that corresponds to A number, presented in Table 6. The A number corresponds to the time in which the 1^{st} device waits for the 2^{nd} to complete its FT. Once the 2^{nd} device completes this activity, both devices are transported to a pre-specified location for their R (T4). The 2^{nd} device starts its repair when the repair of the 1^{st} device ends. The repair path presents the highest average time for both cases. This is due to the time taken for the transition of the device(s) from the FT to R. The same procedure was applied for the other paths and the additional average times in the case of two devices in the process are shown in Table 6 for each path. The f

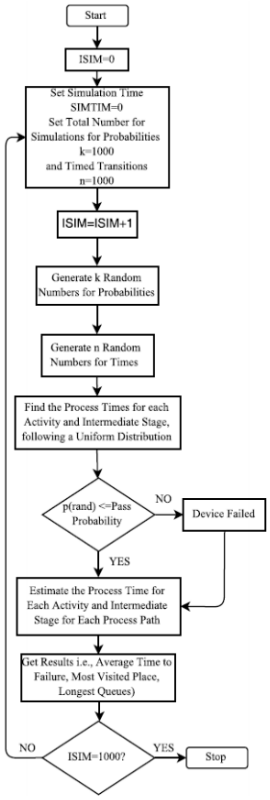

Figure 4 Flowchart for IT Process Simulation.

Table 3 Average Times of Timed Transitions in the Repair Process Path.

Timed Transitions ID	T1	Ti1	T2	Ti3	T4	Ti4	T3
Average Time (seconds)	127.77	74.951	7.510	1039.79	119.19	47536.15	569.968

numbers in the C&D and S&S paths correspond to the time that the 2^{nd} device waits for the 1^{st} to end with the T7 and T8 respectively. According to the A and f numbers, placed in Table 6, the longest queue has been identified in the repair path, since the sum of A and T4 is higher than the corresponding combinations of the other paths. The C&D and all the S&S paths have similar numbers of queues, since the f numbers are very close and T7 is equal to T8. Based on the simulation only the 2.69% of the devices can be repaired, the 48.9% undergoes the C&D and the rest of the devices lead to the S&S paths.

4 Conclusions

In this paper a process of recycling IT assets was investigated by developing and subsequently simulating a PN for the process. The simulation results have given a clear understanding of the working way for the six different paths of which the process consists and also main limitations and potential modifications have been identified in order to enhance the process performance. A main limiting factor of the process is the long time needed for the Repair path to be completed. This happens because the Repair activity takes place in a different physical location from the other activities. Additional observation for all the process paths is that the mobile phones spend the most time within the intermediate times rather than

Table 4 Most Common Failed Nodes in Each Process Path.

Node ID (Place)	Pi4	Pi3	Pi11	Pi5	Pi5	Pi8
No. of Failures	596.0384	312.0666	688.059	595.9192	596.0097	489.1843

Table 5 Paths in the IT Process.

Path	Activities
1	AT-VI(P)-FT(F-No-Erase)-R
2	AT-VI(P)-FT(P)-DE(P)-C&D
3	AT-VI(F)-S&S
4	AT-VI(P)-FT(F-Erase)-DE(P)-S&S
5	AT-VI(P)-FT(F-Erase)-DE(F)-S&S
6	AT-VI(P)-FT(P)-DE(F)-S&S

Table 6 Overall Average Times of Each Path for 1 and 2 Devises and Additional Time Required in the Case of 2 Devices.

Path	Avg. Time(1 phone)(secs)	Avg. Time(2 phones)(secs)	Avg. Additional Time(secs)	
1	4.87E+04	4.94E+04	A=126.7688,	T4
2	7763.5347	7941.2791	f=82.4264,	T7
3	2206.6686	2331.9461	f=82.4199,	T8
4	7761.9031	7940.2581	f=82.3125,	T8
5	7762.7325	7941.008	f=82.4972,	T8
6	7763.5059	7941.1462	f=82.9941,	T8

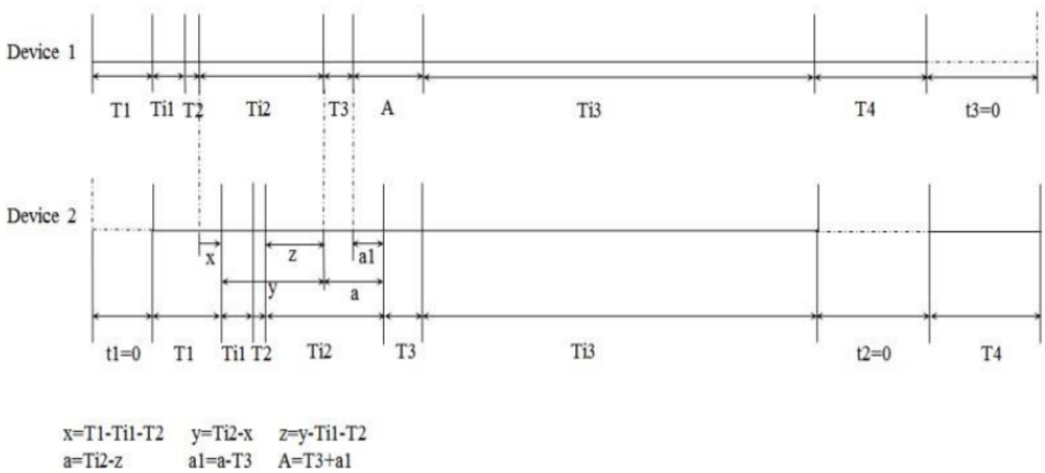

Figure 5 Estimation of Repair Average Time for 2 Devices.

within the activities. So, some potential ways of improving the efficiency of the process could be the duplication of the activities, increase of the manpower and decrease of the intermediate times, for example the devices could be repaired at the same location, or the transportation of the devices could be arranged to take place after the end of the workers' shifts and hence the intermediate time between the Functional Test and Repair would not affect the whole time process. Another recommendation for the process enhancement is the creation of standards for the activities to accept multiple devices simultaneously, so that this can save not only time and money but also can increase the reputation of the company.

The future work involves; (a) applying the proposed process to a variety of possible scenarios, i.e. more than two devices in the process. This will allow the user to identify where the technical investment should be made or how to rearrange the process in order to improve its performance and (b) developing an algorithm to generate automatically the PN model for the IT process from the system, or process description. This algorithm will then be generalised for any complex system, or process. The manual PN developed in this work will be used to validate the algorithm. The automated generation of the PN will contribute to the literature by enabling different designs to be investigated precisely, easily and quickly; hence allowing optimal designs to be determined.

References

1 Mark Lanus, Liang Yin, and Kishor S Trivedi. Hierarchical composition and aggregation of state-based availability and performability models. *IEEE Transactions on Reliability*, 52(1):44–52, 2003.

2 Chris Ling. An introduction to petri nets. [Power Point Presentation], 2001.

3 Christian Mahulea, Michaela H. Matcovschi, and Octavian Pastravanu. Home page of the petri net toolbox, 2003.

4 Seyed H. Nejad-Hosseinian. *Automatic generation of generalized event sequence diagrams for guiding simulation based dynamic probabilistic risk assessment of complex systems*. PhD thesis, University of Maryland, College Park, Department of Mechanical Engineering, 2007.

5 Carl A. Petri. *Communication with automata*. PhD thesis, Universität Hamburg, 1966.

6 Jiacun Wang. Petri nets for dynamic event-driven system modeling. *Handbook of Dynamic System Modeling*, pages 1–17, 2007.

7 Valérie Zille, Christophe Bérenguer, Antoine Grall, and Antoine Despujols. Simulation of maintained multicomponent systems for dependability assessment. In *Simulation Methods for Reliability and Availability of Complex Systems*, pages 253–272. Springer, 2010.